Markovian Discriminative Modeling for Dialog State Tracking

Hang Ren, Weiqun Xu, Yonghong Yan
The Key Laboratory of Speech Acoustics and Content Understanding
Institute of Acoustics, Chinese Academy of Sciences
21 North 4th Ring West Road, Beijing, China, 100190
{renhang, xuweiqun, yanyonghong}@hccl.ioa.ac.cn

Abstract

Discriminative dialog state tracking has become a hot topic in dialog research community recently. Compared to generative approach, it has the advantage of being able to handle arbitrary dependent features, which is very appealing. In this paper, we present our approach to the DSTC2 challenge. We propose to use discriminative Markovian models as a natural enhancement to the stationary discriminative models. The Markovian structure allows the incorporation of 'transitional' features, which can lead to more efficiency and flexibility in tracking user goal changes. Results on the DSTC2 dataset show considerable improvements over the baseline, and the effects of the Markovian dependency is tested empirically.

1 Introduction

Spoken dialog systems (SDS) have become much more popular these days, but still far from wide adoption. One of the most outstanding problems that affect user experience in an SDS is due to automatic speech recognition (ASR) and spoken language understanding (SLU) errors. While the advancement of ASR technology has a positive effect on SDS, it is possible to improve the SDS user experience by designing a module which explicitly handles ASR and SLU errors. With accurately estimated dialog state, the dialog manager could select more effective and flexible dialog actions, resulting in shorter dialogs and higher dialog success rate. Dialog state tracking is the task of identifying the correct dialog state (user action, user goal, etc.) from ASR and SLU outputs in the presence of errors. Commercial dialog systems these days usually use simple dialog state tracking strategies that only consider the most probable SLU output. Previous research shows that several errors in dialog state tracking can be rectified by considering the full N-best results from the ASR and SLU components (Williams, 2012). Thus it is very important to develop robust and practical dialog state tracking models.

In statistical dialog state tracking, models can be roughly divided into two major classes, i.e. generative and discriminative. Generative (Bayesian) dialog tracking models are prevalent in early studies due to its close relationship with the POMDP dialog management model (Young et al., 2013). Generative models generally use Dynamic Bayesian Networks to model the observation probability $P(O_t|S_t)$ and transition probability $P(S_t|S_{t-1})$, where O_t and S_t are observations and dialog state at turn t. In a discriminative model, the conditional probability $P(S_t|O_1^t)$ is modeled directly, where O_1^t is all the observations from turn 1 to t. One problem with the generative models is that the independent assumptions are always not realistic. For example, N-best hypotheses are often assumed independent of each other, which is flawed in realistic scenarios (Williams, 2012). Furthermore, it is intrinsically difficult for generative models to handle overlapping features, which prevents model designers from incorporating arbitrarily large feature set. Discriminative model does not suffer from the above problems as there is no need to make any assumptions about the probabilistic dependencies of the features. As a result. it is potentially able to handle much larger feature sets and to make more accurate predictions (Bohus and Rudnicky, 2006). Discriminative models also tend to be more data-driven, unlike generative models in which many sub-models parameters are heuristically tuned.

2 DSTC1 revisited

The first Dialog State Tracking Challenge (DSTC1) for the first time provided a common test bed for various state tracking methods, and

Proceedings of the SIGDIAL 2014 Conference, pages 327–331,
Philadelphia, U.S.A., 18-20 June 2014.

several participants employed various discriminative models in the challenge. DSTC1 provided real user dialog corpora in the domain of bus route service to evaluate performance of various state tracking methods. In DSTC1 there are 9 teams with 27 submissions, where discriminative, generative and rule-based models are used in the challenge. Maximum entropy models (Lee and Eskenazi, 2013), conditional random fields (Lee, 2013) and neural networks (Henderson et al., 2013) are the most frequently used discriminative models, which gave competitive results on several metrics. It has been empirically analyzed that discriminative methods are especially advantageous when the ASR/SLU confidence scores are poorly estimated (Williams et al., 2013).

3 Discriminative modeling in dialog state tracking

In the design of a slot-filling or task-oriented dialog systems, dialog state tracking can be considered as a classification problem, i.e. assigning predefined *values* to a fixed number of *slots*. One major problem in the formulation is that in complex dialog scenarios the number of classes tends to be very big, resulting in extremely sparse training instances for each class. This sparsity affects the classification performance. A large prediction domain also leads to computation inefficiency which makes the model less practical. Usually we could focus only on the *on-list* hypotheses, which are the hypotheses appeared in the SLU results, and all the other values in the slot value set are grouped into a meta category `Other`. It is similar to the *partition* concept in HIS (Young et al., 2010), and by doing this we could reduce the number of classes to a reasonable size. We use $\mathcal{Y}_t$ to denote the prediction domain at turn t. Although the number of classes is reduced by focusing on the dynamically generated $\mathcal{Y}_t$, some classes will still suffer from the lack of training instances, and what is even worse is that a large portion of the classes will not have any training data, since in practical SDS deployment it is hard to collect a large dialog corpus. To handle the data sparseness problem, parameters are often shared across different slots, or even data sets, and by doing this the model complexity could be effectively controlled and the overfitting problem would be alleviated. Williams proposed to use various techniques from multi-domain learning to improve model performance (Williams, 2013), which could be taken as another way of parameter sharing.

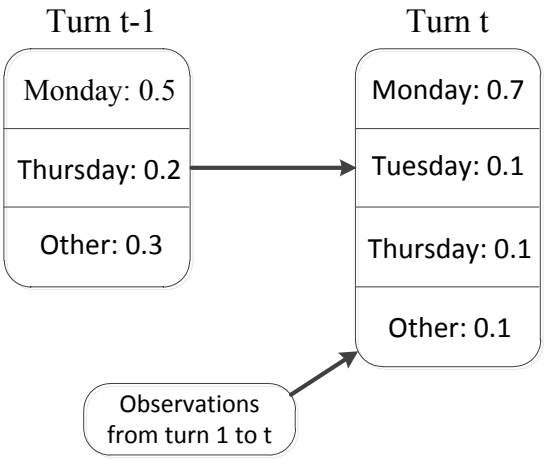

Figure 1: Markovian discriminative model dependency diagram. In this figure the dialog state is simplified to a single slot variable: *date*, the domain of the slot typically increases as dialog continues, which includes all the slot values appeared as SLU results. As indicated by the arrows, S_t depends on S_{t-1} and O_1^t. In stationary discriminative model, there's no dependency between adjacent turns indicated by the upper arrow.

3.1 Markovian discriminative model

A dialog can be naturally seen as a temporal sequence involving a user and an agent, where strong dependencies exist between adjacent turns. In typical task-oriented dialogs, users often change their goals when their original object cannot be satisfied. Even when the true user goal stays constant in a dialog session, the agent's perception of it will tend to evolve and be more accurate as the conversation proceeds, and thus the dialog state will often change. The states at adjacent turns are statistically correlated, and therefore it is important to leverage this natural temporal relationship in tracking dialog state. We enhance the stationary discriminative model in a similar way as described in (McCallum et al., 2000), by assuming Markovian dependency between adjacent turns.

Thus, the original probability $P(S_t|O_1^t)$ can be factored into the following form:

$$P(S_t|O_1^t) = \sum_{S_{t-1}\in\mathbf{S}} P(S_t|O_1^t, S_{t-1})P(S_{t-1}|O_1^{t-1}) \quad (1)$$

The graphical model is shown is figure 1. Unlike stationary discriminative models,

we model the *conditional transition* probability $P(S_t|O_1^t, S_{t-1})$ instead of $P(S_t|O_1^t)$ and the dialog state is updated according to equation 1 at each turn. The feature functions in the structured model can depend on the state of the previous turn, which we call *transitional* features.

It is worth noting that stationary discriminative model can include features built from dialog history (Metallinou et al., 2013). The major difference in utilizing this information from our approach is that by explicitly assuming the Markovian dependency, the structured model is able to exploit the whole probabilistic dialog state distribution of the previous turn. The previous dialog state S_{t-1} is inferred from previous dialog history O_1^{t-1}, which contains higher level hypotheses than the raw history features. Apart from that, the structured model can also use any stationary features built from O_1^t, which makes the stationary model a special case of the structured one.

3.2 Neural network classifier

We use the family of multi-layer neural networks to model the transition probability $P(S_t|O_1^{t-1}, S_{t-1})$. To allow for the use of the dynamic prediction domain, we utilize a forward network structure similar to (Henderson et al., 2013). Feature vectors for each class in $\mathcal{Y}_t$ are fed into the model and forwarded through several hidden layers for non-linear transformation in the hope that deeper layers may form higher abstraction of the raw inputs. The parameter vectors for each class are shared. For each feature vector the model generates a real score. The scores for all the classes in $\mathcal{Y}_t$ are then normalized using a softmax function resulting in valid probabilities.

$$y_i = W_{l-1} \cdot g_{l-1}(\ldots g_1(W_1 \cdot X_i)\ldots) \quad (2)$$

$$P_{\mathbf{Y}} = Softmax(y_1, \ldots, y_{|\mathcal{Y}_t|}) \quad (3)$$

where g_1 to g_{l-1} are sigmoid functions, W_i is the weight matrix for linear transformation at layer i and $X_i = f(O_1^t, y_i)$ is the feature vector for class i. We also test maximum entropy models, which can be seen as a simple neural network without hidden layers:

$$P(\mathbf{Y} = y|O_1^t) = \frac{e^{\lambda \cdot f(O_1^t, y)}}{\sum_{y \in \mathcal{Y}} e^{\lambda \cdot f(O_1^t, y)}} \quad (4)$$

4 DSTC2 challenge

DSTC2 is the second round of Dialog State Tracking Challenge, and it provides dialog corpora collected from real human-machine dialogs in a restaurant domain. The corpora are split into labeled training and development sets and unlabeled test set. Test sets are collected from a SDS different from the training and development set to reflect the mismatch in real deployment. Unlike DSTC1, the user goal often changes in DSTC2 when the condition specified by the user cannot be met. For evaluation DSTC2 defined a number of metrics among which several featured metrics are selected. Besides tracking *user goals* (the values of each slot), two additional states *method* and *requested slots* are also defined, which track the method to query and the slots requested by users respectively. Further details about DSTC2 could be found in (Henderson et al., 2014).

5 Feature set

We briefly describe the feature set used in our system. We only use the live SLU information provided by the organizers, and no extra external data is used. The features used can be divided into two classes.

stationary features which only depend on the observations and the class (slot value) predicted at current turn in the form of $f(y_t, O_t)$.

transitional features that can also depends on the predicted class at the previous turn in the form of $f(y_t, y_{t-1}, O_t)$.

Stationary features include:

- SLU Scores: confidence scores of the current prediction binned into boolean values, raw scores are also added as real features.
- SLU Status: whether the prediction is denied, informed and confirmed in the current turn.
- Dialog history: whether the prediction has been denied, informed and confirmed in all the dialog turns until the current one.
- User/system action: The most probable user action and the machine action in the current turn.

The transitional features are as follows:

- Transition1: whether the predictions in the previous and the current turn are the same.

Name	Model Class	Hidden layers
Entry1	MEMM	–
Entry2	Structured NN	[50]
Entry3	Structured NN	[50, 30]
MLP	Stationary NN	[50, 30]

Table 1: Configurations of models. The model MLP uses the same structure as Entry3, but without the transitional features described in section 5. Number in brackets denotes the number of units used in each hidden layers.

- Transition2: joint feature of Transition1 in conjunction with the machine action in current turn, i.e. for each machine cation, Transision1 is replicated and only the one corresponding to the machine action at current turn is activated.

Transitional features are specific to Markovian models while stationary features can be used in any discriminative models.

6 Model training

Markovian models in various forms are tested to find the most appropriate structure for the task. Models for 'method' and 'state' are built separately using similar structured models.

When using the maximum entropy model to build the conditional probability, the Markovian model is equivalent to the maximum-entropy Markov model (MEMM) model introduced in (McCallum et al., 2000). More sophisticated neural networks with different configurations are used to fit the model to more complex patterns in the input features. In tracking the state 'goal', the joint distribution of slots is built assuming different slots are independent of each other. From the perspective of practical implementation, one advantage of the simpler MEMM model is that the training objective is convex. Thus the optimization routine is guaranteed to find the global optimum, while neural networks with hidden layers always have many local optima which require careful initialization of the parameters. LBFGS (Liu and Nocedal, 1989) is used in optimizing the batch log-likelihood objective and L1 and L2 regularizers are used to penalize the model from overfitting. We train the model on the training set, the development set is used for model selection and models produced at each training iteration are evaluated.

State	Tracker	ACC	L2	CA05
Goal	Baseline	0.619	0.738	0.000
	Entry1	0.707	0.447	**0.223**
	Entry2	0.713	**0.437**	0.207
	Entry3	**0.718**	0.461	0.100
	MLP	0.713	0.448	0.128
Method	Baseline	0.875	0.217	0.000
	Entry1	0.865	0.228	0.199
	Entry2	0.871	0.211	**0.290**
	Entry3	0.871	0.210	0.287
	MLP	**0.946**	**0.092**	0.000
Requested	Baseline	0.884	0.196	0.000
	Entry1	0.932	0.118	0.057
	Entry2	0.947	0.093	0.218
	Entry3	**0.951**	**0.085**	0.225
	MLP	0.863	0.231	**0.291**

Table 2: Evaluation results on the DSTC2 test set. ACC stands for accuracy, L2 measures the Euclidean distance between the predicted distribution and the ground truth vector with only the correct hypothesis set to 1. CA05 is the correct acceptance rate when false acceptance rate is 5%. Details of the metrics can be found in (Henderson et al., 2014). Except L2, the larger the scores, the better the performance.

In DSTC2 we submitted 3 trackers, an additional tracker without the transitional features is trained afterwards for comparison. Configurations of the models are described in table 1.

7 Experiments and part of the results

Featured metrics on the test set are shown in table 2. By most metrics our models are superior to the simple baseline. Especially in tracking user goals which is the most important state to track in DSTC2, the discriminative trackers show considerable performance gain. Judging from the performance of Entry1 to Entry3, we can conclude that the more complex 2-layer neural networks have better performance. Markovian neural networks can fit to the training instances with much more flexibility than the simple MEMM model. We have also trained a standard multi-layer neural network (MLP) model by disabling all the transitional features. By comparing the model 'Entry 3' and 'MLP', which share the same network structure, we explicitly test the effect of the Markovian structure. On the state 'goal' and 'requested', the Markovian model shows better tracking accu-

racies, which means that the Markovian structure has a positive effect on fitting the target. But in tracking the state 'method', the MLP model has the best performance among all the models compared. Thus although the log-likelihood increases considerably on the training set by adding the transitional features, the overfiting to the training set is more serious in tracking 'method'.

8 Conclusion

We described the models used in the DSTC2 challenge. We proposed a novel approach to enhancing the model capability of stationary discriminative models in dialog state tracking by assuming Markovian dependencies between adjacent turns. The results showed better performance than the simple baseline which uses the most probable hypothesis, and we empirically compared the models with and without the Markovian dependency. In future work, more discriminative models in different forms could be compared to evaluate their capability, and the effects of the Markovian structure and transitional features needs to be further studied.

Acknowledgments

We would like to thank the DSTC committee for their great efforts in organizing the challenge. We also thank the anonymous reviewers for their constructive comments.

This work is partially supported by the National Natural Science Foundation of China (Nos. 10925419, 90920302, 61072124, 11074275, 11161140319, 91120001), the Strategic Priority Research Program of the Chinese Academy of Sciences (Grant Nos. XDA06030100, XDA06030500), the National 863 Program (No. 2012AA012503) and the CAS Priority Deployment Project (No. KGZD-EW-103-2).

References

Dan Bohus and Alex Rudnicky. 2006. A k-hypotheses+ other belief updating model. In *Proc. of the AAAI Workshop on Statistical and Empirical Methods in Spoken Dialogue Systems*.

Matthew Henderson, Blaise Thomson, and Steve Young. 2013. Deep neural network approach for the dialog state tracking challenge. In *Proceedings of the SIGDIAL 2013 Conference*, pages 467–471, Metz, France, August. Association for Computational Linguistics.

Matthew Henderson, Blaise Thomson, and Jason Williams. 2014. The second dialog state tracking challenge. In *Proceedings of the SIGDIAL 2014 Conference*, Baltimore, U.S.A., June.

Sungjin Lee and Maxine Eskenazi. 2013. Recipe for building robust spoken dialog state trackers: Dialog state tracking challenge system description. In *Proceedings of the SIGDIAL 2013 Conference*, pages 414–422, Metz, France, August. Association for Computational Linguistics.

Sungjin Lee. 2013. Structured discriminative model for dialog state tracking. In *Proceedings of the SIGDIAL 2013 Conference*, pages 442–451, Metz, France, August. Association for Computational Linguistics.

Dong C Liu and Jorge Nocedal. 1989. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1-3):503–528.

Andrew McCallum, Dayne Freitag, and Fernando C. N. Pereira. 2000. Maximum entropy markov models for information extraction and segmentation. In Pat Langley, editor, *ICML*, pages 591–598. Morgan Kaufmann.

Angeliki Metallinou, Dan Bohus, and Jason Williams. 2013. Discriminative state tracking for spoken dialog systems. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 466–475, Sofia, Bulgaria, August. Association for Computational Linguistics.

Jason Williams, Antoine Raux, Deepak Ramachandran, and Alan Black. 2013. The dialog state tracking challenge. In *Proceedings of the SIGDIAL 2013 Conference*, page 404–413, Metz, France, August. Association for Computational Linguistics.

Jason Williams. 2012. A critical analysis of two statistical spoken dialog systems in public use. In *2012 IEEE Spoken Language Technology Workshop (SLT)*, pages 55–60.

Jason Williams. 2013. Multi-domain learning and generalization in dialog state tracking. In *Proceedings of the SIGDIAL 2013 Conference*, pages 433–441, Metz, France, August. Association for Computational Linguistics.

Steve Young, Milica Gašić, Simon Keizer, François Mairesse, Jost Schatzmann, Blaise Thomson, and Kai Yu. 2010. The hidden information state model: A practical framework for POMDP-based spoken dialogue management. *Computer Speech & Language*, 24(2):150–174.

Steve Young, Milica Gašić, Blaise Thomson, and Jason D Williams. 2013. Pomdp-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179.